AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) An error correcting code decoding device based on Message-Passing decoding on a Low-Density Parity-Check Code, whose parity-checkmatrix consists of sub-matrices of a Kronecker product of two permutation matrices, comprising:

a plurality of memory means for storing a received value and a message generated during said decoding;

a plurality of variable node function means which perform variable node processing in said decoding;

a plurality of check node function means which perform check node processing in said decoding;

a plurality of address generation means for generating an address of said memory means on the nasis of the first permutation matrix is said sub-matrix of a Kronecker product; and

a plurality of shuffle network means for determining a connection between said variable node function means on the basis of the second permutation matrix in said sub-matrix of a Kronecker product~~and said check node function means~~;

wherein said check node functions(s) means perform(s) check node processing sequentially on a unit of said second permutation matrix,

and said variable node functions generate messages in accordance with said current check node processing.

~~wherein said address generation means generates said address on the basis of a plurality of permutations and~~

~~each of said shuffle network means being connected to some of said variable node function means,~~

~~said connection being determined on the basis of a plurality of permutations,~~

~~a change of said permutations in said address generation means and a change of said permutations in said shuffle network means being performed in the same cycle in a decoding process.~~

2. (currently amended) The error correcting code decoding device according to claim 1, wherein said address generation means singly generate[[s]] an address for all of said memory means; and

wherein said shuffle network means [[is]] are singly connected to all of said variable node function means.

3. (currently amended) The error correcting code decoding device according to claim 1, wherein said memory means store[[s]] said message with a sum thereof.

4. (currently amended) The error correcting code decoding device according to claim 1, wherein said address generation means [[is]] are provided as a counter.

5. (currently amended) The error correcting code decoding device according to claim 1, wherein a second permutation by said shuffle network means is determined on [[the]] a basis of a Galois field calculation.

6. (currently amended) The error correcting code decoding device according to claim 1, wherein said decoding corrects a message of an output from said check node function means by multiplying [[it]] the output by a coefficient less than 1 on [[the]] a basis of [[the]] a min-sum algorithm.

7. (currently amended) The error correcting code decoding device according to claim 1, wherein in said decoding, said check node function means hold[[s the]] a first minimum value of [[the]] an absolute value of an input message and an index thereof, and [[the]] a second minimum value of the input message and information regarding whether the input message is positive or negative on [[the]] a basis of [[the]] a min-sum algorithm.

8. (original) The error correcting code decoding device according to claim 1, wherein decoding on a different code is dealt with by changing only said address generation means.

9. (currently amended) The error correcting code decoding device according to claim 1, wherein decoding on a[[n]] uniform Low-Density Parity-Check Code is implemented by providing a function to always send a message that [[the]] an output has a codeword bit with an extremely high probability of 0 to a set of said variable node function means corresponding to one of said address generation means and said shuffle network means.

10. (currently amended) A program to cause a computer to perform decoding on [[the]] a basis of Message-Passing decoding on a Low-Density Parity-Check Code, wherein said program causes said computer to function as:

a plurality of variable node function means in said decoding;

a plurality of check node function means in said decoding;

address generation means for generating addresses of a plurality of memory means that store a received value and a message generated during said decoding, on [[the]] a basis of a plurality permutations; and

shuffle network means for determining a connection between variable node function means and check node function means on [[the]] a basis of a permutation changed in [[the]] a same cycle as that of said address generation means.

11. (currently amended) The program according to claim 10, wherein said memory means store[[s]] said message with a sum thereof.

12. (currently amended) The program according to claim 10, wherein said program determines a permutation in said shuffle network means on [[the]] a basis of a Galois field calculation.

13. (currently amended) The program according to claim 10, wherein said decoding corrects a message of an output from said check node function means by multiplying [[it]] the output by a coefficient less than 1 on [[the]] a basis of [[the]] a min-sum algorithm.

14. (currently amended) The program according to claim 10, wherein in said decoding, said check node function means hold[[s the]] a first minimum value of [[the]] an absolute value of an input message and an index thereof, and [[the]] a second minimum value of the input message and information regarding whether the input message is positive or negative on [[the]] a basis of [[the]] a min-sum algorithm.

15. (currently amended) The program according to claim 10, wherein decoding on a different code is dealt with by changing only [[the]] a function of said address generation means.

16. (currently amended) The program according to claim 10, wherein decoding on a[[n]] uniform Low-Density Parity-Check Code is implemented by providing a function to always send a message that [[the]] an output has a codeword bit with an extremely high probability of 0 to a set of said variable node function means corresponding to one of said address generation means and said shuffle network means.

17. (currently amended) An error correcting code decoding method on [[the]] a basis of Message-Passing decoding on a Low-Density Parity-Check Code, comprising ~~the steps of~~:

generating an address of a memory storing a received value and a message generated during said decoding on [[the]] a basis of a plurality of permutations; and

connecting a plurality of variable node functions in said decoding and a plurality of check node functions in said decoding on [[the]] a basis of a permutation changed in [[the]] a same cycle as that of [[said]] an address generation means.

18. (currently amended) The error correcting code decoding method according to claim 17, wherein said memory stores [[a]] said message with a sum thereof.

19. (currently amended) The error correcting code decoding method according to claim 17, wherein a connection between a variable node function and a check node function is determined on [[the]] a basis of a Galois field calculation.

20. (currently amended) The error correcting code decoding method according to claim 17, wherein said decoding corrects a message of an output from said check node functions by multiplying [[it]] the output by a coefficient less than 1 on [[the]] a basis of [[the]] a min-sum algorithm.

21. (currently amended) The error correcting code decoding method according to claim 17, wherein in said decoding, said check node functions hold[[s the]] a first minimum value of [[the]] an absolute value of an input message and an index thereof, and [[the]] a second minimum value of the input message and information regarding whether the input message is positive or negative on [[the]] a basis of [[the]] a min-sum algorithm.

22. (original) The error correcting code decoding method according to claim 17, wherein decoding on a different code is dealt with by changing address generation in memory.